UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/723,863	11/26/2003	Jason P. Chalecki	MS1-1699US	9696

22801 7590 02/08/2007

LEE & HAYES PLLC
421 W RIVERSIDE AVENUE SUITE 500
SPOKANE, WA 99201

EXAMINER	
TRAN, QUOC A	
ART UNIT	**PAPER NUMBER**
2176	

SHORTENED STATUTORY PERIOD OF RESPONSE	NOTIFICATION DATE	DELIVERY MODE
3 MONTHS	02/08/2007	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Notice of this Office communication was sent electronically on the above-indicated "Notification Date" and has a shortened statutory period for reply of 3 MONTHS from 02/08/2007.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

lhptoms@leehayes.com

Office Action Summary	**Application No.** 10/723,863	**Applicant(s)** CHALECKI ET AL.	
	Examiner Tran A. Quoc	**Art Unit** 2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 November 2006*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-45* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-45* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date *11-26-2006*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. This is a Non-Final rejection in response to amendment filed on 11-24-2006.

2. Effective filing date is 11-26-2003, which claims CIP of 10/610, 504 filed 06-30-2003.

3. Claims 1-45 are pending.

4. Claims 1, 21, 26, 33, 34, and 41 are independent claims.

Response to Argument

5. Applicant's Remarks filed on 11-24-2006 with respect to claim 1-45 have been considered but are moot in view of the new ground(s) of rejection. This office action is a Non-Final Rejection in order to give the applicant sufficient opportunity to response to the new line of rejection.

Claim Objection

6. Regarding claims 14 and 43 the phrase "**a certain range**" and "**the likelihood**" indefinite (Applicant's "In The Claims" pages 8 and 20), renders the claimed indefinite, because Applicant's invention specification merely discloses " script-based custom the validation rule, and real-time validation rule.... nodes governing by other nodes. Thus the real-time validation tool can validate data from a data-entry filed" (Applicant's invention specification para [0122]), and the validation tool item 136 can build validation rules from schema contain logic that governs a structure data file. This logic **sets forth the bounds** of what are the nodes...," (Applicant's invention specification para 0124). It is unclear what Applicant's intended the metes and bounds of the claims are, thereby rendering the scope of the claim(s) unascertainable (an open-ended ranges). Regarding claim 13 the phase "one or ore" is assumed to be a typo error of "one or more".

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> ***(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.***

Claims 1-45 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Bradley et al.** US20050194219A1- filed 04-15-2002 (hereinafter Bradley), in view of **Ricker et al.** US20020049790A1 filed 7-2-2001 (hereinafter Ricker).

Regarding independent claims 1, 21, 26, 33, 37 and 41, Bradley teaches **receiving an instruction to open an eXtensible Markup Language (XML) document; opening the XML document with the solution, wherein the XML document can be inferred from the XML schema and portions of the XML document are logically coupled with fragments of the XML schema; and executing the presentation application to render a HTML electronic form containing data-entry fields associated with the coupled portions.** Specifically, Bradley discloses a method to create a form definition template for collecting information from a user on a browser platform using XML (Bradley para 16, fig. 1), and Bradley also discloses in Fig. 2 a browser, and a designer window display using one or more of standard user interface, an object tool box, toolbars, and property browser (Bradley para 68, fig. 2), also Bradley discloses a user interface using DHTML, HTML, and java for automatically detecting the appropriate browser environment, then transforming and delivering the user-interface in the appropriate browser language, wherein the server exact the field information, such type and the positioning, and the boilerplate information and coverts it to a format that best suit the target browser on the client desktop (Bradley para 40, and 48), and Bradley discloses the uses of XML Forms Architecture ("XFA" is an open, public specification that defines how a form will appear and act in an XML environment. See http://www.w3.org/1999/05/XFA/xfa-template.html for details of XFA). Separating its data elements from the details of its graphic presentation, XFA assumes no proprietary data schema, which means that an enterprise can use the system for a broad range of e-process operations. Because XFA works with a large number of browsers or computer platforms, an enterprise can confidently treat all users in the electronic domain the same (Bradley para 54-55).

In the broadest reasonable interpretation, Examiner interprets the claimed **logically coupled with fragments of the XML schema** as equivalent to exact the field information, such type and the positioning, and the boilerplate information and coverts it to a format that best suit the target browser on the client, and the claimed **XML document with the solution** as equivalent to XML form definition template as taught by Bradley, and because the Applicant's invention specification discloses the electronic from template is involved when a user navigate to an xml document (Applicant's invention specification para 14).

Figure 1

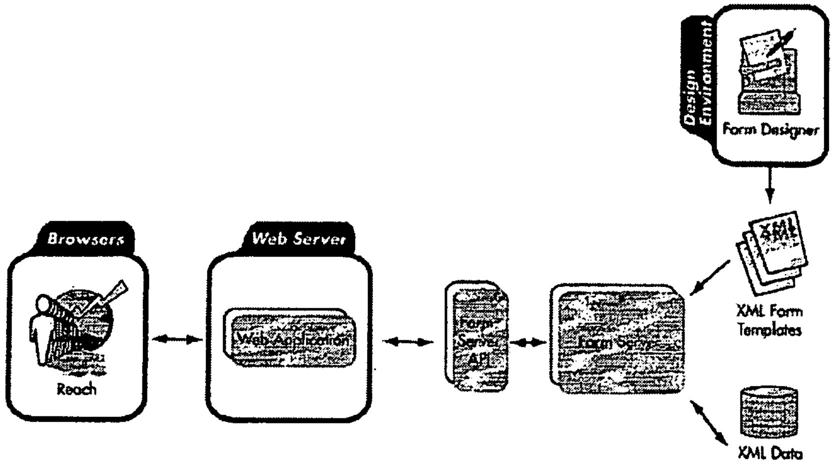
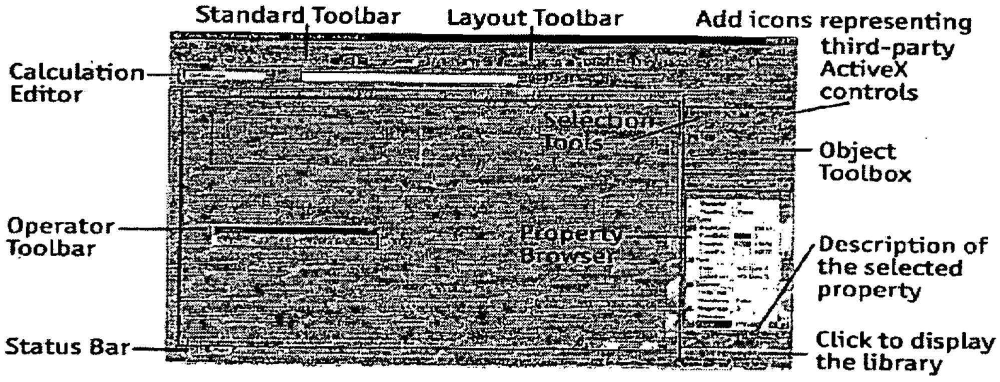
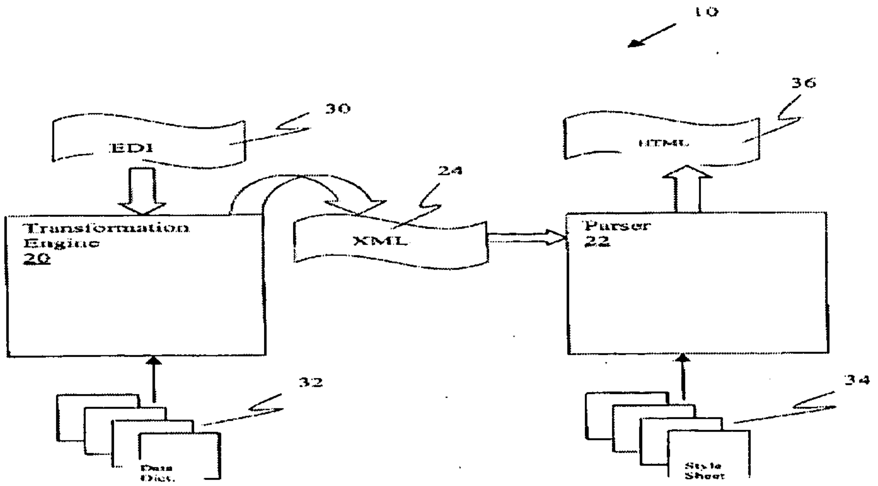


Figure 2



In addition, Bradley discloses the support of SOAP remote function calls, which means that the Form Sever is flexible and exist at either the second or third tier within the architecture. Bradley does not expressly teach, but Ricker teaches **searching the XML document to locate a processing instruction (PI) containing a href attribute that points to a URL; discovering a solution using the URL in the processing instruction (PI).** Specifically, Ricker discloses a method for expressing the content of data interchange format messages, such as Electronic Data Interchange (EDI) documents, in a markup language, such as Extensible Markup Language (XML) (Ricker Abstract). In addition Ricker discloses XML document and can be referenced by multiple parent items, and there can be a separate XML document for each transaction set, segment, and element that is defined by the EDI standards. The reference specifies the URL, or other link or pointer, of the XML file that contains the definition of that segment or element. The URLs are relative to the root directory of data dictionary 32. Example of href specifies the URL shows below: (Ricker para 99 and fig. 1)

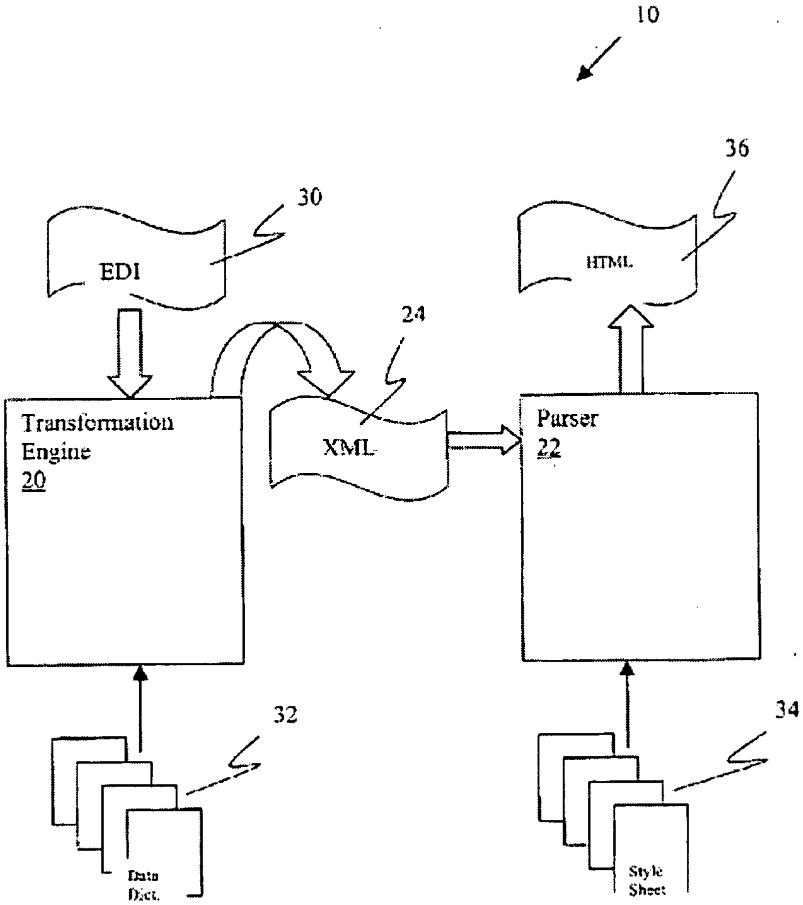
Fig. 1



```
<?xml version=“1.0”?>
<!--Copyright (c) 2001 XMLSolutions Corporation. All rights reserved.-->
<transactionSet code=“850” functionalID=“PO” lang=“EN”>
     <name>Purchase Order</name>
     <version>004040</version>
     <segmentRef code=“ST” req=“M” maxOccurence=“1”
href=“S_ST.xml”>Transaction Set Header</segmentRef>
     <segmentRef code=“BEG” req=“M” maxOccurence=“1”
href=“S_BEG.xml”>Beginning Segment for Purchase
          Order</segmentRef>
     <segmentRef code=“CUR” req=“O” maxOccurence=“1”
href=“S_CUR.xml”>Currency</segmentRef>
     <segmentRef code=“REF” req=“O” maxOccurence=“>1”
href=“S_REF.xml”>Reference Identification.</segmentRef>
     <segmentRef code=“PER” req=“O” maxOccurence=“3”
href=“S_PER.xml”>Administrative Communications
          Contact</segmentRef>
     <segmentRef code=“TAX” req=“O” maxOccurence=“>1”
href=“S_TAX.xml”>Tax Reference</segmentRef>
     <segmentRef code=“FOB” req=“O” maxOccurence=“>1”
href=“S_FOB.xml”>F.O.B. Related Instructions</segmentRef>
     <segmentRef code=“CTP” req=“O” maxOccurence=“>1”
href=“S_CTP.xml”>Pricing Information</segmentRef>
     <segmentRef code=“PAM” req=“O” maxOccurence=“10”
href=“S_PAM.xml”>Period Amount</segmentRef>
     <segmentRef code=“CSH” req=“O” maxOccurence=“5”
href=“S_CSH.xml”>Sales Requirements</segmentRef>
     <segmentRef code=“TC2” req=“O” maxOccurence=“>1”
href=“S_TC2.xml”>Commodity</segmentRef>
     <loop code“SAC” req=“O” maxOccurence=“25”>
          <segmentRef code=“SAC” req=“O” maxOccurence=“1”
     href=“S_SAC.xml”>Service, Promotion, Allowance, or
                         Charge Information</segmentRef>
          <segmentRef code“CUR” req=“O” maxOccurence=“1”
href=“S_CUR.xml”>Currency</segmentRef>
     </loop>
...
</transaction Set>
```

In addition, Bradley does not expressly teach, but Ricker teaches **the solution includes an extensible stylesheet language (XSLT) presentation application and a XML schema, and executing the XSLT presentation application to render a Hypertext Markup Language (HTML).** For example, Ricker discloses Extensible Stylesheet Language for Transformations. XSLT is used to describe and transform a source XML tree into a result tree, which may represent a completely different structure. Transformation options include alternate XML representations, HTML, flat files and PDF (Ricker para 41, and fig. 1 transformation engine item 20).

Fig. 1



It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Bradley to include validating fields of a form using XSL as taught by Ricker, because Bradley and Ricker are from the same field of endeavor of cross–platform xml document authoring user interface tool and logic transformation using XSLT, and providing the benefit of automatic error detection, preventing from entering incorrect data, delaying the process of e-form, and allows uses to customize and simplifying forms design process for faster and make changes quicker, boosting their productivity and their effectiveness as they create (Bradley para 45).

Regarding claims 2, 22, 29, 34, 38 and 42, the rejection of claims 1, 21, 26, 33, 37 and 41 are fully incorporated. In addition, Bradley teaches **executing** the transformation **presentation that are performed by the execution of HTML electronics forms application that is different from the application used to create the HTML electronic form.** Specifically, Bradley discloses a user interface using DHTML, HTML, and java for automatically detecting the appropriate browser environment, then transforming and delivering the user-interface in the appropriate browser language, wherein the server exact the field information, such type and the positioning, and the boilerplate information and coverts it to a format that best suit the target browser on the client desktop.

In addition, Bradley does not expressly teach, but Ricker teaches the **executing the XSLT presentation**. For example, Ricker discloses Extensible Stylesheet Language for Transformations. XSLT is used to describe and transform a source XML tree into a result tree, which may represent a completely different structure. Transformation options include alternate

XML representations, HTML, flat files and PDF (Ricker para 41, and fig. 1 transformation engine item 20).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Bradley to include validating fields of a form using XSL as taught by Ricker, because Bradley and Ricker are from the same field of endeavor of cross–platform xml document authoring user interface tool and logic transformation using XSLT, and providing the benefit of automatic error detection, preventing from entering incorrect data, delaying the process of e-form, and allows uses to customize and simplifying forms design process for faster and make changes quicker, boosting their productivity and their effectiveness as they create (Bradley para 45).

Regarding claim 3, Bradley does not expressly teach, but Ricker teaches **wherein: the executing the XSLT presentation application comprises applying an XSLT stylesheet to the XML document to create the HTML electronic form; and the HTML electronic form includes a plurality of HTML elements corresponding to the data-entry fields.** For example, Ricker discloses Extensible Stylesheet Language for Transformations. XSLT is used to describe and transform a source XML tree into a result tree, which may represent a completely different structure. Transformation options include alternate XML representations, HTML, flat files and PDF document (Ricker para 41, and fig. 1 transformation engine item 20).

In the broadest reasonable interpretation, Examiner interprets the claimed **the HTML electronic form** as equivalent to HTML, flat files and PDF document as taught by Ricker.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Bradley to include validating fields of a form using XSL as taught by Ricker, because

Bradley and Ricker are from the same field of endeavor of cross–platform xml document authoring user interface tool and logic transformation using XSLT, and providing the benefit of automatic error detection, preventing from entering incorrect data, delaying the process of e-form, and allows uses to customize and simplifying forms design process for faster and make changes quicker, boosting their productivity and their effectiveness as they create (Bradley para 45).

Regarding claim 4, Bradley does not expressly teach, but Ricker teaches **searching the XML document to find the first instruction in the XML document that includes a URL.** Specifically, Ricker discloses a method for expressing the content of data interchange format messages, such as Electronic Data Interchange (EDI) documents, in a markup language, such as Extensible Markup Language (XML) (Ricker Abstract). In addition Ricker discloses XML document and can be referenced by multiple parent items, and there can be a separate XML document for each transaction set, segment, and element that is defined by the EDI standards. The reference specifies the URL, or other link or pointer, of the XML file that contains the definition of that segment or element. The URLs are relative to the root directory of data dictionary 32. Example of href specifies the URL shows below: (Ricker para 99)

```
<?xml version=“1.0”?>
<!--Copyright (c) 2001 XMLSolutions Corporation. All rights reserved.-->
<transactionSet code=“850” functionalID=“PO” lang=“EN”>
     <name>Purchase Order</name>
     <version>004040</version>
     <segmentRef code=“ST” req=“M” maxOccurence=“1”
href=“S_ST.xml”>Transaction Set Header</segmentRef>
     <segmentRef code=“BEG” req=“M” maxOccurence=“1”
href=“S_BEG.xml”>Beginning Segment for Purchase
          Order</segmentRef>
     <segmentRef code=“CUR” req=“O” maxOccurence=“1”
href=“S_CUR.xml”>Currency</segmentRef>
     <segmentRef code=“REF” req=“O” maxOccurence=“>1”
href=“S_REF.xml”>Reference Identification.</segmentRef>
     <segmentRef code=“PER” req=“O” maxOccurence=“3”
href=“S_PER.xml”>Administrative Communications
          Contact</segmentRef>
     <segmentRef code=“TAX” req=“O” maxOccurence=“>1”
href=“S_TAX.xml”>Tax Reference</segmentRef>
     <segmentRef code=“FOB” req=“O” maxOccurence=“>1”
href=“S_FOB.xml”>F.O.B. Related Instructions</segmentRef>
     <segmentRef code=“CTP” req=“O” maxOccurence=“>1”
href=“S_CTP.xml”>Pricing Information</segmentRef>
     <segmentRef code=“PAM” req=“O” maxOccurence=“10”
href=“S_PAM.xml”>Period Amount</segmentRef>
     <segmentRef code=“CSH” req=“O” maxOccurence=“5”
href=“S_CSH.xml”>Sales Requirements</segmentRef>
     <segmentRef code=“TC2” req=“O” maxOccurence=“>1”
href=“S_TC2.xml”>Commodity</segmentRef>
     <loop code“SAC” req=“O” maxOccurence=“25”>
          <segmentRef code=“SAC” req=“O” maxOccurence=“1”
     href=“S_SAC.xml”>Service, Promotion, Allowance, or
                    Charge Information</segmentRef>
          <segmentRef code“CUR” req=“O” maxOccurence=“1”
href=“S_CUR.xml”>Currency</segmentRef>
     </loop>
...
</transaction Set>
```

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Bradley to include searching the XML document to find the first instruction in the XML document that includes a URL as taught by Ricker, because Bradley and Ricker are from the same field of endeavor of cross–platform xml document authoring user interface tool and logic transformation using XSLT, and providing the benefit of automatic error detection, preventing from entering incorrect data, delaying the process of e-form, and allows uses to customize and simplifying forms design process for faster and make changes quicker, boosting their productivity and their effectiveness as they create (Bradley para 45).

Regarding claim 5, Bradley teaches **a manifest of all files that can be used for representing the XML document in the HTML electronic form; allowing a user in input data into the data-entry fields; and validating the data that the user inputs into the data-entry fields**. Specifically, Bradley discloses a method to create a form definition template for collecting information from a user on a browser platform using XML (Bradley para 16, fig. 1), and Bradley also discloses in Fig. 2 a browser, and a designer window display using one or more of standard user interface, an object tool box, toolbars, and property browser (Bradley para 68, fig. 2), also Bradley discloses a user interface using DHTML, HTML, and java for automatically detecting the appropriate browser environment, then transforming and delivering the user-interface in the appropriate browser language, wherein the server exact the field information, such type and the positioning, and the boilerplate information and coverts it to a format that best suit the target browser on the client desktop (Bradley para 40, and 48).

In the broadest reasonable interpretation, Examiner interprets the claimed **logically coupled with fragments of the XML schema** as equivalent to exact the field information, such type and the positioning, and the boilerplate information and coverts it to a format that best suit the target browser on the client, and the claimed **a manifest of all files that can be used for representing the XML document in the HTML electronic form** as equivalent to XML form definition template as taught by Bradley, and because the Applicant's invention specification discloses the electronic from template is involved when a user navigate to an xml document. (Applicant's invention specification para 14).

Regarding claim 6, Bradley teaches **the coupled portions contain information setting forth all possible XML documents for the coupled portions.** Specifically Bradley discloses the uses of XML Forms Architecture ("XFA" is an open, public specification that defines how a form will appear and act in an XML environment. See http://www.w3.org/1999/05/XFA/xfa-template.html for details of XFA). Separating its data elements from the details of its graphic presentation, XFA assumes no proprietary data schema, which means that an enterprise can use the system for a broad range of e-process operations. Because XFA works with a large number of browsers or computer platforms, an enterprise can confidently treat all users in the electronic domain the same (Bradley para 54-55).

Regarding claims 7, 23, 27, 35, 39 and 44, Bradley teaches **data-entry fields of the HTML form mapping to corresponding nodes of XML document, and receiving data input by user and outputting data in XML for viewing in HTML forms through data-entry fields; and the method further comprises: outputting data in XML for viewing by the user in the HTML electronic form through the data-entry fields via the mapping of the data-entry fields from corresponding said nodes of the XML document.** Specifically, Bradley discloses a method to create a form definition template for collecting information from a user on a browser platform using XML (Bradley para 16, fig. 1), and Bradley also discloses in Fig. 2 a browser, and a designer window display using one or more of standard user interface, an object tool box, toolbars, and property browser (Bradley para 68, fig. 2), also Bradley discloses "subform area container nodes" in the XFA specification. They allow server side scripts, such as VBScript and JScript, to work with multiple occurrence fields (Bradley para 150-188), wherein HTML pages, Application Server Pages ("ASP"s), ColdFusion pages and others. Using the API, the Web application gathers information from the end user's browser. From a single template, the Server determines the browser type and then transforms the form template, with or without data, into a format that best suits that particular browser type (Bradley para 64). In the broadest reasonable interpretation, Examiner interprets the claimed **HTML form mapping to corresponding nodes of XML document** as equivalent to exact the field information, such type and the positioning, and the boilerplate information and coverts it to a format that best suit the target browser on the client, and the claimed **XML document with the solution** as equivalent to HTML pages, Application Server Pages ("ASP"s), ColdFusion pages and others. Using the API, the Web application gathers information from the end user's browser. From a single template (XML form template (Bradley para 16, fig. 1)), the Server determines the browser type and then transforms the form template as taught by Bradley.

Figure 1

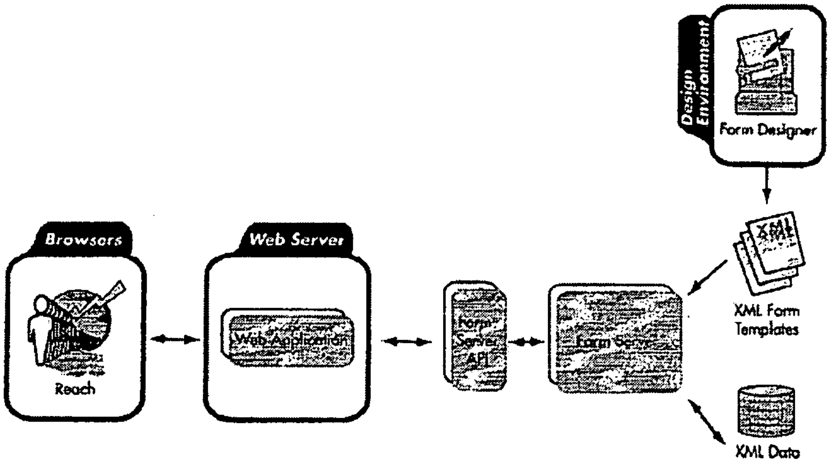
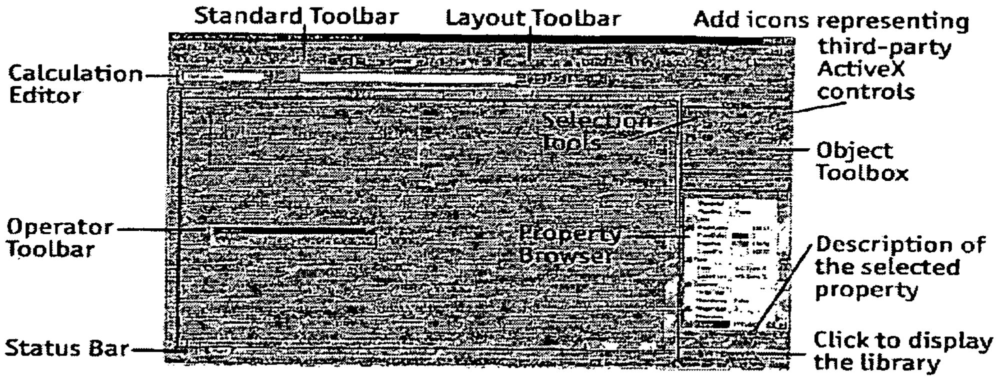


Figure 2

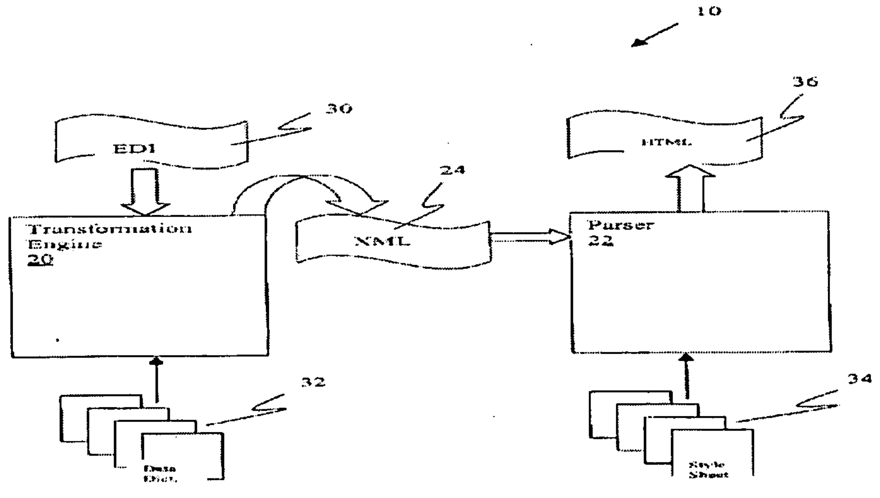


Regarding claim 8, Bradley teaches **the XML schema includes a logic application; and the method further comprises: executing the logic application to perform a validation to determine if the data received by input from the user is valid or invalid; and when the validation determines that the data received by input from the user is invalid, outputting a dialog box bearing indicia informing the user that the data input is invalid.** For example, Bradley discloses the methods that change the state of an object, wherein scripts can invoke methods to perform validations based on a specific event, and return a specific value to a field (Bradley para 115-118).

Regarding claim 9, Bradley teaches **the validation is performed on the data received by input from the user into each said data-entry field with a validation rule; the logic application comprises a plurality of the validation rules for: a corresponding plurality of the nodes in the XML document; and a corresponding plurality of the data-entry fields; the validation uses each said validation rule to: determine if the data received by input from the user into a corresponding said data-entry field is valid or invalid; and require the user to correct any data input into the corresponding said data-entry field that the validation determines to be invalid.** Bradley discloses, "subform area container nodes" in the XFA specification. They allow server side scripts, such as VBScript and JScript, to work with multiple occurrence fields (Bradley para 150-188), also Bradley discloses the methods that change the state of an object, wherein scripts can invoke methods to perform validations based on a specific event, and return a specific value to a field (Bradley para 115-118), furthermore Bradley discloses employing the Designer's WYSIWYG graphical design tools for user interfaces and automatic error checking to prevent respondents from entering incorrect data and delaying the processing of their e-forms (Bradley para 39).

Regarding claim 10, Bradley does not expressly teach, but Ricker teaches **validation rule has an identity that is selected from the group consisting of: the identity is based on a part of a schema governing a corresponding said node; the identity is written in script and associated with a corresponding said 20 node; and the identity is written in a declarative syntax and associated with a corresponding said node.** Specifically, Ricker discloses a method for expressing the content of data interchange format messages, such as Electronic Data Interchange (EDI) documents, in a markup language, such as Extensible Markup Language (XML) (Ricker Abstract). In addition Ricker discloses XML document and can be referenced by multiple parent items, and there can be a separate XML document for each transaction set, segment, and element that is defined by the EDI standards. The reference specifies the URL, or other link or pointer, of the XML file that contains the definition of that segment or element. The URLs are relative to the root directory of data dictionary 32. Example of href specifies the URL shows below: (Ricker para 99 and fig. 1)

Fig. 1



```
<?xml version=“1.0”?>
<!--Copyright (c) 2001 XMLSolutions Corporation. All rights reserved.-->
<transactionSet code=“850” functionalID=“PO” lang=“EN”>
     <name>Purchase Order</name>
     <version>004040</version>
     <segmentRef code=“ST” req=“M” maxOccurence=“1”
href=“S_ST.xml”>Transaction Set Header</segmentRef>
     <segmentRef code=“BEG” req=“M” maxOccurence=“1”
href=“S_BEG.xml”>Beginning Segment for Purchase
          Order</segmentRef>
     <segmentRef code=“CUR” req=“O” maxOccurence=“1”
href=“S_CUR.xml”>Currency</segmentRef>
     <segmentRef code=“REF” req=“O” maxOccurence=“>1”
href=“S_REF.xml”>Reference Identification.</segmentRef>
     <segmentRef code=“PER” req=“O” maxOccurence=“3”
href=“S_PER.xml”>Administrative Communications
          Contact</segmentRef>
     <segmentRef code=“TAX” req=“O” maxOccurence=“>1”
href=“S_TAX.xml”>Tax Reference</segmentRef>
     <segmentRef code=“FOB” req=“O” maxOccurence=“>1”
href=“S_FOB.xml”>F.O.B. Related Instructions</segmentRef>
     <segmentRef code=“CTP” req=“O” maxOccurence=“>1”
href=“S_CTP.xml”>Pricing Information</segmentRef>
     <segmentRef code=“PAM” req=“O” maxOccurence=“10”
href=“S_PAM.xml”>Period Amount</segmentRef>
     <segmentRef code=“CSH” req=“O” maxOccurence=“5”
href=“S_CSH.xml”>Sales Requirements</segmentRef>
     <segmentRef code=“TC2” req=“O” maxOccurence=“>1”
href=“S_TC2.xml”>Commodity</segmentRef>
     <loop code“SAC” req=“O” maxOccurence=“25”>
          <segmentRef code=“SAC” req=“O” maxOccurence=“1”
     href=“S_SAC.xml”>Service, Promotion, Allowance, or
                         Charge Information</segmentRef>
          <segmentRef code“CUR” req=“O” maxOccurence=“1”
href=“S_CUR.xml”>Currency</segmentRef>
     </loop>
...
</transaction Set>
```

In the broadest reasonable interpretation, Examiner interprets the claimed **the identity is based on a part of a schema** as equivalent to exact the field information, such type and the positioning, and the boilerplate information and coverts it to a format that best suit the target browser on the client as taught by Bradley.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Bradley to include validation rule has an identity that is selected from the group consisting of: the identity is based on a part of a schema governing a corresponding said node as taught by Ricker, because Bradley and Ricker are from the same field of endeavor of cross–platform xml document authoring user interface tool and logic transformation using XSLT, and providing the benefit of automatic error detection, preventing from entering incorrect data, delaying the process of e-form, and allows uses to customize and simplifying forms design process for faster and make changes quicker, boosting their productivity and their effectiveness as they create (Bradley para 45).

Regarding claim 11, Bradley teaches **wherein: each said validation rule includes an alert area display; and the validation further comprises: using one said validation rule to determine that the data received by input from the user into a corresponding said data-entry field is invalid; and outputting the corresponding alert area display so as to be associated with the corresponding said data-entry field.** For example, Bradley discloses the methods that change the state of an object, wherein scripts can invoke methods to perform validations based on a specific event, and return a specific value to a field (Bradley para 115-118), furthermore Bradley discloses employing the Designer's WYSIWYG graphical design tools for user interfaces and automatic error checking to prevent respondents from entering incorrect data and delaying the processing of their e-forms (Bradley para 39).

Regarding claim 12, Bradley teaches **wherein when the alert area display is output, the output includes one or more characteristics selected from the group consisting of: graphics surrounding the corresponding said data-entry field; the alert area display surrounds the corresponding said data-entry field; the alert area display includes graphics containing a red, dashed-lined box; the alert area display includes graphics highlighting the data in the corresponding said data-entry field; the alert area display surrounds the corresponding said data-entry field and includes the graphics containing a squiggly line beneath the data in the corresponding said data-entry field; the alert area display includes text containing information about the invalid data in the corresponding said data-entry field; the alert area display includes text containing corresponding said data-entry field; and the alert area display includes a pop-up window.** For example, Bradley discloses the methods that change the state of an object, wherein scripts can invoke methods to perform validations based on a specific event, and return a specific value to a field (Bradley para 115-118), furthermore Bradley discloses employing the Designer's WYSIWYG graphical design tools for user interfaces and automatic error checking to prevent respondents from entering incorrect data and delaying the processing of their e-forms (Bradley para 39).

Regarding claim 13, Bradley does not expressly teach, but Ricker teaches **wherein each said node has one or ore of the validation rules associated therewith.** Specifically, Bradley discloses the form objects and pages. These are known as "subform area container nodes" in the XFA specification (Bradley para 150), and Bradley discloses intelligent templates for user interfaces for validate a user's data entry before processing by performing calculations, accessing databases or enforcing business rules on field-level data. Whenever data is submitted to the server, the Server merges the data it has received into the template and executes the business logic contained in the template. The resulting data is then returned to the browser (Bradley para 49).

Regarding claim 14, Bradley teaches **wherein one said validation rule includes a requirement that is selected from the group consisting of: the data received by input from the user into a corresponding said data-entry field is to be within a certain range; the data received by input from the user into a corresponding said data-entry field is to be within a certain range of text or numerals for a setting of one or more bounds of the certain range; the data received by input from the user into a corresponding said data-entry field is to be numerical; the data received by input from the user into a corresponding said data-entry field is to be textual; and the data received by input from the user into a corresponding said data-entry field that references another said node in the data file.** Specifically, Bradley discloses intelligent templates for user interfaces. An enterprise can validate a user's data entry before processing by performing calculations, accessing databases or enforcing business rules on field-level data. Whenever data is submitted to the server, the Server merges the data it has received into the template and executes the business logic contained in the template. The resulting data is then returned to the browser (Bradley para 49).

In the broadest reasonable interpretation, Examiner interprets the claimed **data-entry field is to be within a certain range** as equivalent to exact the field information, such type and the positioning, and the boilerplate information and coverts it to a format that best suit the target browser on the client, and the claimed **node** as equivalent to processing by performing calculations user's data entry, accessing databases or enforcing business rules on field-level data as taught by Bradley, and because Applicant's invention specification merely discloses " script-based custom the validation rule, and real-time validation rule.... nodes governing by other nodes. Thus the real-time validation tool can validate data from a data-entry filed" (Applicant's invention specification para [0122]), and the validation tool item 136 can build validation rules from schema contain logic that governs a structure data file. This logic **sets forth the bounds** of what are the nodes...," (Applicant's invention specification para 0124).

Regarding claim 15, Bradley teaches **wherein the plurality of the validation rules re associated by mapping to the corresponding plurality of the nodes in the data file.** For example, Bradley discloses form templates contain fillable and non-fillable objects. These objects are used for data entry and are typically mapped to the XML data file (Bradley para 219), and Bradley discloses the methods that change the state of an object, wherein scripts can invoke methods to perform validations based on a specific event, and return a specific value to a field (Bradley para 115-118).

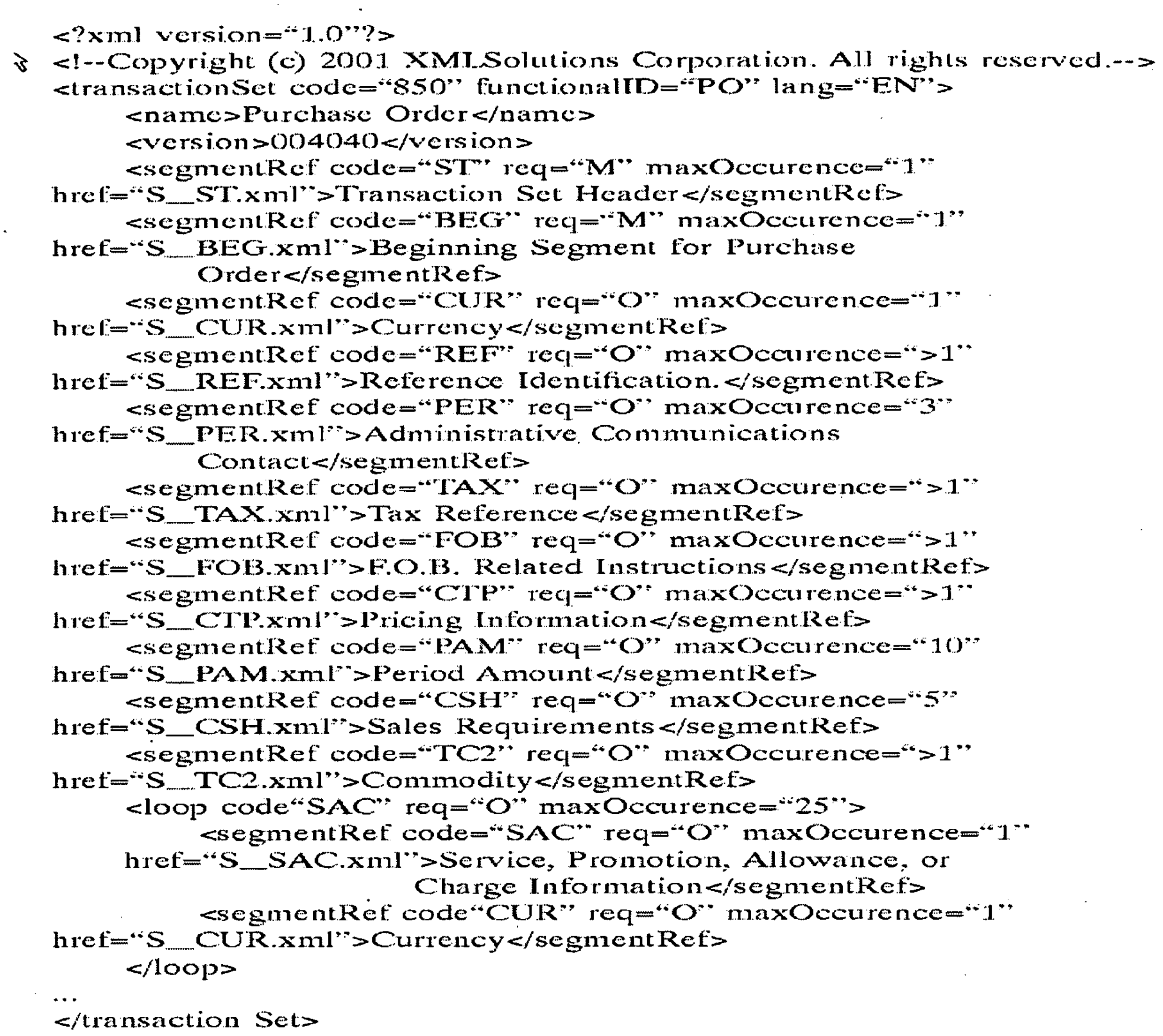
Regarding claim 16, Bradley teaches **validation rule is associated by mapping to a corresponding said data-entry field by use an entity selected from the group consisting of an XPath expression, a declarative syntax, and an entity that is script-based.** For example, Bradley discloses the methods that change the state of an object, wherein scripts can invoke methods to perform validations based on a specific event, and return a specific value to a field (Bradley para 115-118).

Regarding claim 17, Bradley teaches **wherein the script-based validation rule maps to a corresponding said node with an entity selected from the group consisting of: an XPath expression; an event handler; an event handier that determines when a real-time validation tool uses the script-based validation rule; an event handler that determines when a real-time validation tool uses the script-based validation rule before data received for the node is held by the data file; and an event handler that determines when a real-time validation tool uses the script-based validation rule after data received for the node is held by the data file**. For example, Bradley discloses the methods that change the state of an object, wherein scripts can invoke methods to perform validations based on a specific event, and return a specific value to a field (Bradley para 115-118). In the broadest reasonable interpretation, Examiner interprets the claimed **an event handler** as equivalent to the state of an object as taught by Bradley.

Regarding claim 18, Bradley teaches **wherein each said validation rule includes an alert area display; and how the alert area display is to appear when output**. For example, Bradley discloses the methods that change the state of an object, wherein scripts can invoke methods to perform validations based on a specific event, and return a specific value to a field (Bradley para 115-118), furthermore Bradley discloses employing the Designer's WYSIWYG graphical design tools for user interfaces and automatic error checking to prevent respondents from entering incorrect data and delaying the processing of their e-forms (Bradley para 39).

Regarding claims 19, 24, 28, and 45, Bradley does not expressly teach, but Ricker teaches **wherein the PI includes a character string of "mso-infoPathSolution".** Specifically, Ricker discloses XML document and can be referenced by multiple parent items, and there can be a separate XML document for each transaction set, segment, and element that is defined by the EDI standards. The reference specifies the URL, or other link or pointer, of the XML file that contains the definition of that segment or element. The URLs are relative to the root directory of data dictionary 32. Example of href specifies the URL shows below: (Ricker para 99 and fig. 1).

In the broadest reasonable interpretation, Examiner interprets the claimed **a character string of "mso-infoPathSolution"** as equivalent to specifies the URL, or other link or pointer, of the XML file that contains the definition of that segment or element. The URLs are relative to the root directory of data dictionary item 32 as taught by Bradley.



```
<?xml version=“1.0”?>
<!--Copyright (c) 2001 XMLSolutions Corporation. All rights reserved.-->
<transactionSet code=“850” functionalID=“PO” lang=“EN”>
     <name>Purchase Order</name>
     <version>004040</version>
     <segmentRef code=“ST” req=“M” maxOccurence=“1”
href=“S__ST.xml”>Transaction Set Header</segmentRef>
     <segmentRef code=“BEG” req=“M” maxOccurence=“1”
href=“S__BEG.xml”>Beginning Segment for Purchase
          Order</segmentRef>
     <segmentRef code=“CUR” req=“O” maxOccurence=“1”
href=“S__CUR.xml”>Currency</segmentRef>
     <segmentRef code=“REF” req=“O” maxOccurence=“>1”
href=“S__REF.xml”>Reference Identification.</segmentRef>
     <segmentRef code=“PER” req=“O” maxOccurence=“3”
href=“S__PER.xml”>Administrative Communications
          Contact</segmentRef>
     <segmentRef code=“TAX” req=“O” maxOccurence=“>1”
href=“S__TAX.xml”>Tax Reference</segmentRef>
     <segmentRef code=“FOB” req=“O” maxOccurence=“>1”
href=“S__FOB.xml”>F.O.B. Related Instructions</segmentRef>
     <segmentRef code=“CTP” req=“O” maxOccurence=“>1”
href=“S__CTP.xml”>Pricing Information</segmentRef>
     <segmentRef code=“PAM” req=“O” maxOccurence=“10”
href=“S__PAM.xml”>Period Amount</segmentRef>
     <segmentRef code=“CSH” req=“O” maxOccurence=“5”
href=“S__CSH.xml”>Sales Requirements</segmentRef>
     <segmentRef code=“TC2” req=“O” maxOccurence=“>1”
href=“S__TC2.xml”>Commodity</segmentRef>
     <loop code“SAC” req=“O” maxOccurence=“25”>
          <segmentRef code=“SAC” req=“O” maxOccurence=“1”
     href=“S__SAC.xml”>Service, Promotion, Allowance, or
                         Charge Information</segmentRef>
          <segmentRef code“CUR” req=“O” maxOccurence=“1”
href=“S__CUR.xml”>Currency</segmentRef>
     </loop>
...
</transaction Set>
```

Regarding claims 20, 25, 32, 36, 40, the rejection of claims 1, 21, 26, 33, and 37 are fully incorporated. In addition, a computer readable medium, when executed by a computer embodied therein for performance the method of claims 1, 21, 26, 33, and 37 (Bradley para 19).

Regarding claim 30, the rejection of claim 1 is fully incorporated. In addition, Bradley teaches **the assessment of the likelihood exceeds the threshold when the PI is the first PI in the XML document that contains a URL.** Specifically, Bradley discloses the intelligent templates for user interfaces. An enterprise can validate a user's data entry before processing by performing calculations, accessing databases or enforcing business rules on field-level data. Whenever data is submitted to the server, the Server merges the data it has received into the template and executes the business logic contained in the template. The resulting data is then returned to the browser (Bradley para 49), also Bradley discloses a path and file name or a Universal Resource Locator (URL) to specify the graphic file to use, or embed the graphic in the form (Bradley para 88). In the broadest reasonable interpretation, Examiner interprets the claimed **the likelihood exceeds the threshold** as equivalent to the intelligent templates for user interfaces. An enterprise can validate a user's data entry before processing by performing calculations, accessing databases or enforcing business rules on field-level data as taught by Bradley.

Regarding claim 31, Bradley teaches **the assessment of the likelihood exceeds the threshold when the one of a URL is a URL having a path with a suffix that is selected from the group consisting of .xsf or .xsn.** Specifically, Bradley discloses the intelligent templates for user interfaces. An enterprise can validate a user's data entry before processing by performing calculations, accessing databases or enforcing business rules on field-level data. Whenever data is submitted to the server, the Server merges the data it has received into the template and executes the business logic contained in the template. The resulting data is then returned to the browser (Bradley para 49), also Bradley discloses a path and file name or a Universal Resource Locator (URL) to specify the graphic file to use, or embed the graphic in the form (Bradley para 88). In addition, Bradley discloses the relationship between the XML form template file (.xft) and the XML form data file (.xfd) (Bradley para 191).

In the broadest reasonable interpretation, Examiner interprets the claimed **the likelihood exceeds the threshold** as equivalent to the intelligent templates for user interfaces. An enterprise can validate a user's data entry before processing by performing calculations, accessing databases or enforcing business rules on field-level data, and the claimed **a path with a suffix that is selected from the group consisting of .xsf or .xsn** as equivalent to XML form template file (.xft) and the XML form data file (.xfd) as taught by Bradley.

Regarding claim 43, the rejection of claims 1, and 30-31 are fully incorporated. In addition, Bradley teaches **examining the name of a URL or an URN in the PI to access the likelihood tat the PI includes a solution identifier for the solution, and a combination of foregoing**. For example, Bradley discloses the intelligent templates for user interfaces. An enterprise can validate a user's data entry before processing by performing calculations, accessing databases or enforcing business rules on field-level data. Whenever data is submitted to the server, the Server merges the data it has received into the template and executes the business logic contained in the template. The resulting data is then returned to the browser (Bradley para 49), also Bradley discloses a path and file name or a Universal Resource Locator (URL) to specify the graphic file to use, or embed the graphic in the form (Bradley para 88). In addition, Bradley discloses the relationship between the XML form template file (.xft) and the XML form data file (.xfd) (Bradley para 191).

In the broadest reasonable interpretation, Examiner interprets the claimed **the likelihood exceeds the threshold** as equivalent to the intelligent templates for user interfaces. An enterprise can validate a user's data entry before processing by performing calculations, accessing databases or enforcing business rules on field-level data, as taught by Bradley.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Quoc A. Tran whose telephone number is 571-272-8664. The examiner can normally be reached on 9AM - 5PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Herndon R. Heather can be reached on 571-272-4136. The fax phone number for the organization where this application or proceeding is assigned is (571)-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Quoc A. Tran,
GAU 2176,
Feb. 05, 2007

H. Herndon

Heather R. Herndon
Supervisory Patent Examiner
Technology Center 2100